SPECIFICATION AMENDMENTS

Please amend the title as follows:

Method for Selective Erasing and Parallel Programming/Verifying of Cell Blocks in a Flash EEprom System

Please amend the Abstract as follows:

A system of Flash EEprom memory chips with controlling circuits serves as non-volatile memory such as that provided by magnetic disk drives. A chunk of user data is programmed into a group of memory cells in parallel, the programming of individual memory cells being terminated when they are determined to have reached desired threshold level ranges while the programming of other memory cells continues. ~~Improvements~~ Other improvements include selective multiple sector erase, in which any combinations of Flash sectors may be erased together. Selective sectors among the selected combination may also be de-selected during the erase operation. Another improvement is the ability to remap and replace defective cells with substitute cells. The remapping is performed automatically as soon as a defective cell is detected. When the number of defects in a Flash sector becomes large, the whole sector is remapped. Yet another improvement is the use of a write cache to reduce the number of writes to the Flash EEprom memory, thereby minimizing the stress to the device from undergoing too many write/erase cycling.

Please amend the Specification as follows, with reference to the substitute specification filed April 28, 2005:

[0001] This is a continuation of patent application serial no. 09/129,675, filed August 5, 1998, which is a continuation of patent application serial no. 08/771,708, filed December 20, 1996, now patent no. 5,991,517, which is a continuation of patent application serial no. 08/174,768, filed December 29, 1993, now patent no. 5,602,987, which in turn is a continuation of patent application serial no. 07/963,838, filed October 20, 1992, now patent no. 5,297,148, which in turn is a division of patent application serial no. 07/337,566, filed April 13, 1989, now abandoned.

[0004] EEprom's and Flash EEprom's are also solid state memory devices. Moreover, they are nonvolatile, and retain their memory even after power is shut down. However, conventional Flash EEprom's have a limited lifetime in terms of the number of write (or program)/erase cycles they can endure. Typically the devices are rendered unreliable after [[102]] 10^2 to [[103]] 10^3 write/erase cycles. Traditionally, they are typically used in applications where semi-permanent storage of data or program is required but with a limited need for reprogramming.

[0034] Fig. 3B shows details of a typical register used to select a sector for erase as shown in Fig. [[2A]] 3A;

[0041] Figure 10 is a view of the structure of Figure 9 taken across section [[2-2]] 10-10 thereof;

[0068] Referring to Figure 1B, the controller 31 is preferably formed primarily on a single integrated circuit chip. It is connected to the system address and data bus 39, part of the system bus [[33]] 23, as well as being connected to system control lines 41, which include interrupt, read, write and other usual computer system control lines.

[0069] The EEprom array 33 includes a number of EEprom integrated circuit chips 43, 45, 47, etc. Each includes a respective chip select and enable line 49, 51 and 53 from interface circuits 40. The interface circuits 40 also act to interface between the serial data lines 35, 37 and a circuit [[55.]] 57. Memory location addresses and data being written into or read from the EEprom chips 43, 45, 47, etc. are communicated from a bus 55, through logic and register circuits 57 and thence by another bus 59 to each of the memory chips 43, 45, 47 etc.

[0082] Optimized erase implementations have been disclosed in two copending U.S. patent applications. They are copending U.S. patent applications, Serial No. 07/204,175, filed June 8, 1988, by Dr. Eliyahou Harari, now patent no. 5,095,344, and one entitled "Multi-State EEprom Read and Write Circuits and Techniques," Serial No. 07/337,579, filed April 13, 1989, now abandoned, by Sanjay Mehrotra and Dr. Eliyahou Harari. The disclosures of the two applications are hereby ~~incorporate~~ incorporated by reference. The Flash EEprom cells are erased by applying a pulse of erasing voltage followed by a read to verify if the cells are erased to the "erased" state. If not, further pulsing and verifying are repeated until the cells are verified to be erased. By erasing in this controlled manner, the cells are not subject to over-erasure which tends to age the EEprom device prematurely as well as make the cells harder to program.

[0084] Returning to figure 4(4), after all sectors intended for erase have been tagged, the controller initiates an erase cycle to erase the group of tagged sectors. In figure 4(5), the controller shifts in a global command called Enable Erase into each Flash EEprom chip that is to perform an erase. This is followed in figure 4(5) by the controller raising of the erase voltage line (Ve) to a specified value for a specified duration. The controller will lower this voltage at the end of the erase duration time. In figure 4(6), the controller will then do a read verify sequence on the sectors selected for erase. In figure 4(7), if none of the sectors are verified, the sequences illustrated in figures 4(5)-4(7) are repeated. In figures 4(8) and [[3(9)]] 4(9), if one or more sectors are verified to be erased, they are taken out of the sequence. Referring also to figure 3A, this is achieved by having the controller address each of the verified sectors and clear the associated erase enable registers back to a LOW with a clear enable command in bus 237. The sequences illustrated in figures 4(5)-4(10) are repeated until all the sectors in the group are verified to be erased in figure 4(11). At the completion of the erase cycle, the controller will shift in a No Operation (NOP) command and the global Enable Erase command will be withdrawn as a protection against a false erasure.

[0087] An additional performance capability of the system in the present invention is the ability to issue a reset command to a Flash EEprom chip which will clear all erase enable latches and will prevent any further erase cycles from occurring. This is illustrated in figures [[2A and

2B]] 3A and 3B by the reset signal in the line 261. By doing this in a global way to all the chips, less time will be taken to reset all the erase enable registers.

[0093] The nature of the Flash EEprom device predicates a higher rate of cell failure especially with increasing program/erase cycling. The hard errors that accumulate with use would eventually overwhelm the ECC and render the device unusable. One important feature of the present invention is the ability for the system to correct for hard errors whenever they occur. Defective cells are detected by their failure to program or erase correctly. Also during read operation, defective cells are detected and located by the ECC. As soon as a defective cell is identified, the controller will apply defect mapping to replace the defective cell with a ~~space~~ spare cell located usually within the same sector. This dynamic correction of hard errors, in addition to conventional error correction schemes, significantly prolongs the life of the device.

[0103] A pipeline architecture is employed to provide efficient throughput as the data is gated through the controller from the receiver 515 to the FIFO 519. As each data bit is received from memory the controller is comparing the address of the data being sent (stored in the address generator ~~507~~ 503) against the defect pointer map (stored in the register file 509). If the address is determined to be a bad location, by a match at the output of the comparator 521, the bad bit from the memory received by the receiver 515 is replaced by the good bit for that location. The good bit is obtained from the alternative defects data file 517. This is done by switching the multiplexer 523 to receive the good bit from the alternative defects data file instead of the bad bit from the receiver 515, as the data is sent to the FIFO 519. Once the corrected data is in the FIFO it is ready to be sent to buffer memory or system memory (not shown). The data is sent from the controller's FIFO 519 to the system memory by the controller's DMA controller 507. This controller 507 then requests and gets access to the system bus and puts out an address and gates the data via an output interface 525 out to the system bus. This is done as each byte gets loaded into the FIFO 519. As the corrected data is loaded into the FIFO it will also be gated into an ECC hardware 527 where the data file will be acted on by the ECC.

[0109] After the bytes for a write cycle have been loaded into the selected memory device, the controller issues a program command to the memory device and initiate a write cycle. Optimized implementations of write operation for Flash EEprom device have been disclosed in two previously cited co-pending U.S. patent applications, Serial No. 07/204,175, now patent no. 5,095,344, and one entitled "Multi-State EEprom Read and Write Circuits and Techniques," Serial No. 07/337,579, filed April 13, 1989, now abandoned. Relevant portions of the disclosures are hereby incorporated by reference. Briefly, during the write cycle, the controller applies a pulse of programming (or writing) voltages. This is followed by a verify read to determine if all the bits have been programmed properly. If the bits did not verify, the controller repeats the program/verify cycle until all bits are correctly programmed.

[0112] In addition, the collection of bits that was flagged as defective and were saved in the alternative defects data file ~~516~~ 517 is then written in memory at the alternative defects data locations (see figure 5), thereby saving the good bit values to be used on a subsequent read. Once these data groups are written and verified, the sector write is considered completed.

[0120] In the present invention, a system of Flash EEprom is used to provide non-volatile memory in place of traditional system memories such as disk storage. However, Flash EEprom memory is subject to wearing out by excessive program/erase cycles. Even with the improved Flash EEprom memory device as disclosed in co-pending U.S. patent applications, Serial No. 07/204,175, now patent no. 5,095,344, and one entitled "Multi-state EEprom Read and Write Circuits and Techniques," by Sanjay Mehrotra and Dr. Eliyahou Harari, Serial No. 07/337,579, filed April 13, 1989, now abandoned, the endurance limit is approximately 10^6 program/erase cycles. In a ten-year projected life time of the device, this translates to a limit of one program/erase cycle per 5 minutes. This may be marginal in normal computer usage.

[0123] Figure 8 illustrates schematically a cache system 701 as part of the controller, according to the present invention. On one hand the cache system 701 is connected to the Flash EEprom memory array 33. On the other hand it is connected to the microprocessor system (not shown) through a host interface 703. The cache system 701 has two memories. One is a cache memory 705 for temporarily holding write data files. The other is a tag memory 709 for storing relevant information about the data files held in the cache memory 705. A memory timing/control circuit 713 controls the writing of data files from the cache memory 705 to the Flash EEprom memory 33. The memory control circuit 713 is responsive to the information stored in the tag memory as well as a power sensing input 715 [[with]] which is connected through the host interface 703 via a line 717 to the power supply of the microprocessor system. A power loss in the microprocessor system will be sensed by the memory control circuit 713 which will then down load all the data files in the volatile cache memory 705 to the non-volatile Flash EEprom memory 33.

[0144] The memory cell is programmed by transferring electrons from the substrate 1015 to a floating gate, such as the floating gate 1025 of the memory cell 1013. The charge on the floating gate 1025 is increased by electrons traveling across the dielectric from a heavily p-doped region 1043 and onto the floating gate. Charge is removed from the floating gate through the dielectric between it and the erase gates 1029 and 1031. This preferred EEprom structure, and a process for manufacturing it, are described in detail in copending patent application Serial No. 07/323,779 of Jack H. Yuan and Eliyahou Harari, filed March 15, 1989, now patent no. 5,070,032, which is expressly incorporated herein by reference.

[0147] The various aspects of the present invention are typically applied to an array of Flash EEprom cells in an integrated circuit chip. Figure 12 illustrates schematically an array of individually addressable EEprom cells 1060. Each cell is equivalent to the one shown in Figure 11, having a control gate, source and drain, and an erase gate. The plurality of individual memory cells are organized in rows and columns. Each cell is addressed by selectively energizing its row and column simultaneously. A column 1062, for example, includes a first memory cell 1063, an adjacent second memory cell 1065, and so forth. A second column 1072 includes memory cells 1073, 1075, and so forth. Cells 1063 and 1073 are located in a row 1076, cells 1065 and [[1071]] 1075 in another, adjacent row, and so forth.

[0156] In a typical operation of the EEprom chip 1130, the controller 1140 will send a serial stream of signals to the chip 1130 via serial in line 1251. The signals, containing control, data, address and timing information, will be sorted out by the serial protocol logic 1170. In appropriate time sequence, the logic 1170 outputs various control signals 1257 to control the various circuits on the chip 1130. It also sends an address via the internal address bus [[111]] 1111 to connect the addressed cell to voltages put out from the controller. In the meantime, if the operation is programming, the data is staged for programming the addressed cell by being sent via a serial data line 1259 to a set of read/program latches and shift registers 1190.

[0159] Copending patent application Serial No. 07/204,175, now patent no. 5,095,344, discloses an EEprom cell with memory states defined within a maximized window of threshold voltage V_{T1}. The full threshold voltage window includes the negative region of the threshold voltage, in addition to the usual positive region. The increased window provides more memory space to implement multi-state in an EEprom cell.

[0168] The present invention overcomes these problems and presents circuits and techniques to reliably program and read the various states even in a multi-state implementation. The memory state of a cell may be determined by measuring the threshold voltage V_{T1} programmed therein. Alternatively, as set forth in co-pending patent application, Serial No. 07/204,175, now patent no. 5,095,344, the memory state may conveniently be determined by measuring the differing conduction in the source-drain current I_{DS} for the different states. In the 4-state example, figure 15A shows the partition in the threshold voltage window. Figure 15B, on the other hand, illustrates typical values of I_{DS} (solid curves) for the four states as a function of the control gate voltage V_{CG}. With V_{CG} at 5V, the I_{DS} values for each of the four conduction states can be distinguished by sensing with four corresponding current sensing amplifiers in parallel. Associated with each amplifier is a corresponding reference conduction states I_{REF} level (shown as broken curves in figure 16). Just as the breakpoint threshold levels (see figures 14 and 15A) are used to demarcate the different regions in the threshold voltage window, the I_{REF} levels are used to do the same in the corresponding source-drain current window. By comparing with the I_{REF}'s, the conduction state of the memory cell can be determined. Co-pending patent application, Serial No. 07/204,175, now patent no. 5,095,344, proposes using the same sensing amplifiers and I_{REF}'s for both programming and reading. This provides good tracking between the reference levels (broken curves in figure 15B) and the programmed levels (solid curves in figure 15B).

[0182] Figure 20A illustrates one embodiment in which the local reference cells are used directly to read or program/erase verify the sector's memory cells. Thus, during those operations, a parallel pair of switches [[1525]] 1521 is enabled by a READ signal and the sense amplifier 1440 will read the sector's addressed memory cells 1523 with respect to each of the sector's local reference cells 1525. During program/erase verify of the local reference cells (as illustrated in figure 19), another parallel pair of switches 1527 enables reading of the local reference cells 1525 relative to the master reference cells 1529.

[0189] The read circuits and operation described are also employed in the programming and erasing of the memory cells, particularly in the verifying part of the operation. As described previously, programming is performed in small steps, with reading of the state programmed in between to verify if the desired state has been reached. As soon as the programmed state is verified correctly, programming stops. Similarly, erasing is performed in small steps, with reading of the state of erase in between to verify if the "erased" state has been [[reach]] reached. Once the "erased" state is verified correctly, erasing stops.

[0201] Figure 25 shows one embodiment of the program circuit with inhibit 1210 of figure 13 in more detail. The program circuit 1210 comprises N program with inhibit modules such as 1801, 1803. As illustrated in the tables of figures 26 and 27, in order to program the N cells, a voltage V_{PD} must be applied to each of the N cells' drain and a voltage V_{PG} applied to the control gates. Each program module such as 1801 serves to selectively pass V_{PD} on a line 1805 to one of the drains through the one of the N-channel data path 1105. Since V_{PD} is typically about 8V to 9V which is higher than V_{CC}, the latter cannot be used to turn on the transistor switch 1807. Rather the higher voltage V_{CG} (about 12V) is used to enable switch 1807. V_{CG} in line [[1801]] 1809 is itself enabled by an AND gate when both the program control signal PGM in line 1813 is true and the signal in line 1731 is a “0”. Since the signal in line 1731 is from the output of the cell compare module 1701 shown in figure 24, it follows that V_{PD} will be selectively passed onto those cells which are not yet verified. In this way, every time a programming pulse is applied, it is only applied to those cells which have not yet reached their intended states. This selective programming feature is especially necessary in implementing parallel programming and on chip verification in the multi-state case.